Docket No.: 042390.P8629

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Carl M. Ellison Application No.: 09/541,667 Filed: March 31, 2000 For: ATTESTATION KEY MEMORY DEVICE AND BUS	Examiner: Tongoc Tran Art Group: 2134

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicants submit, in triplicate, the following Appeal Brief pursuant to 37 C.F.R. § 1.192 for consideration by the Board of Patent Appeals and Interferences. Applicants also submit herewith our check number 31311 in the amount of $330.00 to cover the cost of filing the opening brief as required by 37 C.F.R. § 1.17(f). Please charge any additional fees or credit any overpayment to our deposit Account No. 02-2666. A duplicate copy of the Fee Transmittal is enclosed for this purpose.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is the assignee, Intel Corporation.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to the appellants, the appellants' legal representative, or assignee, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-80 of the present application are pending and remain rejected. The Applicants hereby appeals the rejection of claims 1-80.

IV. STATUS OF AMENDMENTS

The Examiner issued an Office Action on November 12, 2003. The Applicants filed a response on January 14, 2004. Then. Applicants filed an amendment on April 14, 2004, in response to a Final Office Action issued by the Examiner on March 3, 2004. In response to the April 14, 2004 amendment, the Examiner issued an Advisory Action on May 7, 2004. The Applicants filed a Notice of Appeal from the Advisory Action issued by the Examiner on June 3, 2004.

V. SUMMARY OF INVENTION

One embodiment of the invention is to provide interface to an attestation device in a secure environment. An operating system and a processor may have several levels of hierarchy, referred to as rings, corresponding to various operational modes[1]. A logical operating architecture is an abstraction of the components of an operating system and the processor. The logical operating architecture includes ring-0 10, ring-1 20, ring-2 30, ring-3 40, and a processor nub loader 52. The processor nub loader 52 is an instance of a processor executive (PE) handler. The logical operating architecture 50 has two modes of operation: normal execution mode and isolated execution mode. Each ring in the logical

operating architecture 50 can operate in both modes. The processor nub loader 52 operates only in the isolated execution mode[2].

Ring-0 10 includes two portions: a normal execution Ring-0 11 and an isolated execution Ring-0 15. The normal execution Ring-0 11 includes software modules that are critical for the operating system, usually referred to as kernel. These software modules include primary operating system (e.g., kernel) 12, software drivers 13, and hardware drivers 14. The isolated execution Ring-0 15 includes an operating system (OS) nub 16 and a processor nub 18. The OS nub 16 and the processor nub 18 are instances of an OS executive (OSE) and processor executive (PE), respectively. The OSE and the PE are part of executive entities that operate in a secure environment associated with the isolated area 70 and the isolated execution mode. The processor nub loader 52 is a protected bootstrap loader code held within a chipset in the system and is responsible for loading the processor nub 18 from the processor or chipset into an isolated area[3].

One concept of the isolated execution architecture is the creation of an isolated region in the system memory, referred to as an isolated area, which is protected by both the processor and chipset in the computer system. Access to this isolated region is permitted only from a front side bus (FSB) of the processor, using special bus (e.g., memory read and write) cycles, referred to as isolated read and write cycles. The special bus cycles are also used for snooping. The isolated read and write cycles are issued by the processor executing in an isolated execution mode. The isolated execution mode is initialized using a privileged instruction in the processor, combined with the processor nub loader 52. The processor nub loader 52 verifies and loads a ring-0 nub software module (e.g., processor nub 18) into the isolated area. The processor nub 18 provides hardware-related services for the isolated execution[4].

One task of the processor nub 18 is to verify and load the ring-0 OS nub 16 into the isolated area, and to generate the root of a key hierarchy unique to a combination of the platform, the processor nub 18, and the operating system nub 16. The operating system nub 16 provides links to services in the primary OS 12 (e.g., the unprotected segments of

---

[1] See Specification, page 4, lines 9-16.
[2] See Specification, page 4, lines 24-25; page 5, lines 1-8.
[3] See Specification, page 5, lines 9-20.
[4] See Specification, page 6, lines 1-12.

the operating system), provides page management within the isolated area, and has the responsibility for loading ring-3 application modules 45, including applets 46_1 to 46_K, into protected pages allocated in the isolated area. The operating system nub 16 may also load ring-0 supporting modules[5].

The accessible physical memory 60 includes an isolated area 70 and a non-isolated area 80. The isolated area 70 includes applet pages 72 and nub pages 74. The non-isolated area 80 includes application pages 82 and operating system pages 84. The isolated area 70 is accessible only to elements of the operating system and processor operating in isolated execution mode. The non-isolated area 80 is accessible to all elements of the ring-0 operating system and to the processor[6].

The normal execution ring-0 11 including the primary OS 12, the software drivers 13, and the hardware drivers 14, can access both the OS pages 84 and the application pages 82. The normal execution ring-3, including applications 42_1 to 42_N, can access only to the application pages 82. Both the normal execution ring-0 11 and ring-3 41, however, cannot access the isolated area 70[7].

The isolated execution ring-0 15, including the OS nub 16 and the processor nub 18, can access to both of the isolated area 70, including the applet pages 72 and the nub pages 74, and the non-isolated area 80, including the application pages 82 and the OS pages 84. The isolated execution ring-3 45, including applets 46_1 to 46_K, can access only to the application pages 82 and the applet pages 72. The applets 46_1 to 46_K reside in the isolated area 70[8].

The computer system 100 includes at least a processor 110, a host bus 120, a memory controller hub (MCH) 130, a system memory 140, an input/output controller hub (ICH) 150, a non-volatile memory 110, a token bus 180, and a token 186[9].

The processor 110 includes a normal execution mode 112 and an isolated execution circuit 115. The normal execution mode 112 is the mode in which the processor 110

[5] See Specification, page 6, lines 13-20.
[6] See Specification, page 7, lines 9-14.
[7] See Specification, page 7, lines 15-19.
[8] See Specification, page 7, lines 20-25.
[9] See Specification, page 8, lines 1-9.

operates in a non-secure environment, or a normal environment without the security features provided by the isolated execution mode. The isolated execution circuit 115 provides a mechanism to allow the processor 110 to operate in an isolated execution mode. The isolated execution circuit 115 provides hardware and software support for the isolated execution mode. This support includes configuration for isolated execution, definition of an isolated area, definition (e.g., decoding and execution) of isolated instructions, generation of isolated access bus cycles, and generation of isolated mode interrupts[10].

In addition to normal mode, the host bus 120 provides an isolated access bus mode with corresponding interface signals for memory read and write cycles when the processor 110 is configured in the isolated execution mode. The isolated access bus mode is asserted on memory accesses initiated while the processor 110 is in the isolated execution mode[11].

The MCH 130 provides interface circuits to recognize and service isolated access assertions on memory reference bus cycles, including isolated memory read and write cycles. In addition, the MCH 130 has memory range registers (e.g., base and length registers) to represent the isolated area in the system memory 140. Once configured, the MCH 130 aborts any access to the isolated area that does not have the isolated access bus mode asserted[12].

The system memory 140 includes the accessible physical memory 60 (shown in Figure 1B). The accessible physical memory includes a loaded operating system 142, the isolated area 70, and an isolated control and status space 148. The loaded operating system 142 is the portion of the operating system that is loaded into the system memory 140. The isolated area 70 is the memory area that is defined by the processor 110 when operating in the isolated execution mode. Access to the isolated area 70 is restricted and is enforced by the processor 110 and/or the MCH 130 or other chipset that integrates the isolated area functionalities. The isolated control and status space 148 is an input/output (I/O)-like, independent address space defined by the processor 110 and/or the MCH 130. The isolated control and status space 148 contains mainly the isolated execution control and status

[10] See Specification, page 8, lines 18-25; page 9, lines 1-2.
[11] See Specification, page 9, line 25; page 10, lines 1-9.
[12] See Specification, page 10,lines 10-16.

registers. The isolated control and status space 148 does not overlap any existing address space and is accessed using the isolated bus cycles[13].

The ICH 150 represents a known single point in the system having the isolated execution functionality. The ICH 150 has a number of functionalities that are designed to support the isolated execution mode in addition to the traditional I/O functions. In particular, the ICH 150 includes an isolated bus cycle interface 152, the processor nub loader 52, a digest memory 154, a cryptographic key storage 155, an isolated execution logical processor manager 156, and a token bus interface 159[14].

The isolated bus cycle interface 152 includes circuitry to interface to the isolated bus cycle signals to recognize and service isolated bus cycles, such as the isolated read and write bus cycles. The processor nub loader 52, includes a processor nub loader code and its digest (e.g., hash) value. The processor nub loader 52 is invoked by execution of an appropriate isolated instruction (e.g., Iso_Init) and is transferred to the isolated area 70. From the isolated area 80, the processor nub loader 52 copies the processor nub 18 from the system flash memory (e.g., the processor nub code 18 in non-volatile memory 160) into the isolated area 70, verifies and logs its integrity, and manages a symmetric key used to protect the processor nub's secrets. The digest memory 154, typically implemented in RAM, stores the digest (e.g., hash) values of the loaded processor nub 18, the operating system nub 16, and any other critical modules (e.g., ring-0 modules) loaded into the isolated execution space. The cryptographic key storage 155 holds a symmetric encryption/decryption key that is unique for the platform of the system 100. The isolated execution logical processor manager 156 manages the operation of logical processors operating in isolated execution mode. The token bus interface 159 interfaces to the token bus 180. A combination of the processor nub loader digest, the processor nub digest, the operating system nub digest, and optionally additional digests, represents the overall isolated execution digest, referred to as isolated digest. The isolated digest is a fingerprint identifying the ring-0 code controlling the isolated execution configuration and operation. The isolated digest is used to attest or prove the state of the current isolated execution[15].

---

[13] See Specification, page 10, lines 19-26; page 11, lines 1-8.
[14] See Specification, page 11, lines 9-19.
[15] See Specification, page 11, lines 20-26; page 12, lines 1-21.

The non-volatile memory 160 includes the processor nub 18. The processor nub 18 provides the initial set-up and low-level management of the isolated area 70 (in the system memory 140), including verification, loading, and logging of the operating system nub 16, and the management of the symmetric key used to protect the operating system nub's secrets.

The token bus 180 provides an interface between the ICH 150 and various tokens in the system. A token is a device that performs dedicated input/output functions with security functionalities. A token has characteristics similar to a smart card, including at least one reserved-purpose public/private key pair and the ability to sign data with the private key. Examples of tokens connected to the token bus 180 include a motherboard token 182, a token reader 184, and other portable tokens 186 (e.g., smart card). The token bus interface 159 in the ICH 150 connects through the token bus 180 to the ICH 150 and ensures that when commanded to prove the state of the isolated execution, the corresponding token (e.g., the motherboard token 182, the token 186) signs only valid isolated digest information. For purposes of security, the token should be connected to the digest memory[16].

In an embodiment of the present invention, a technique is provided for remote attestation. The remote attestation is performed by a device operating in a remote manner with respect to the MCH 130 and the ICH 150. Examples of this device include one of the tokens 186. This device is referred to as an attestation key memory (AKM) device. This remote attestation is performed by using a public-private key pair to attest that the isolated execution mode is running with a particular software configuration. The AKM device contains one or more key pair and may be inserted into the platform by the end user needed to perform the attestation[17].

In an embodiment of the present invention, an interface maps a device (e.g., the AKM device) via a bus to an address space of a chipset in a secure environment for an isolated execution mode. The secure environment is associated with an isolated memory area accessible by at least one processor. The at least one processor operates in one of a normal execution mode and the isolated execution mode. A communication storage

[16] See Specification, page 14, lines 3-13.
[17] See Specification, page 14, lines 15-25.

corresponding to the address space allows the device to exchange security information with the at least one processor in the isolated execution mode in a remote attestation[18].

The token bus interface 159 includes an interface 210, a communication storage 220, and a chipset storage 270.

The interface 210 provides an interface between an external device (e.g., the tokens 186 shown in Figure 1C) coupled to the token bus 180 and the chipset (e.g., the ICH 150). The interface 210 includes a decoder 212. The decoder 212 decodes the address space onto the bus 180 so that an access to the chipset is passed to the device. Typically the address space is a subset of the address space of the chipset 150. In addition, the decoder 212 allows the device 186 to access the chipset storage 270[19].

The communication storage 220 is mapped to the address space and allows the device 186 to exchange security information with the chipset 150 or the processor 110. The communication storage 220 includes a configuration storage 230, a status register 240, a command register 250, and an input/output block (IOB) 260. The configuration storage 230 stores configuration information 232. The status register 240 stores device status 242. The command register 250 stores device command 252. The IOB 260 stored input data 262 and output data 264[20].

The chipset storage 270 stores chipset information such as the system digest in the digest memory 154 (Figure 1C). In particular, the chipset storage 270 includes a processor nub loader hash 272, a chipset hash log 274, a software hash 276, and a nonce 278. The processor nub loader hash 272 and the chipset hash log 274 can be read directly by the AKM device 186 and cannot be intercepted by the running software. The software hash 276 and the nonce 278 are provided by the processor nub 18[21].

The configuration storage 230 includes a manufacturer identifier 310, a revision identifier 320, an interface set identifier 330, a static public key 340, and a static key certificate 350. The configuration storage includes a plurality of sub-storages (e.g., public

[18] See Specification, page 15, lines 10-17.
[19] See Specification, page 15, lines 21-24; page 16, lines 1-2; Figure 2, element 210.
[20] See Specification, page 16, lines 3-8; Figure 2, element 220
[21] See Specification, page 16, lines 10-15; Figure 2, element 270.

key storage, key certificate storage, interface set storage, revision storage). Typically, the configuration storage 230 is read-only[22].

The manufacturer identifier 310 identifies the manufacturer of the AKM device 186. The revision identifier 320 provides a revision number of the AKM device 186. The interface set identifier 330 identifies the interface set that is supported by the device 186. The static public key 340 is a public key with a short key identification. The key certificate 350 is a key certificate with a short key identification[23].

The interface set identified by the interface set identifier 330 identifies may include an initialization set 360, an attestation set 370, and a device interface set 380. For a typical remote attestation, the initialization set 360 is needed. The initialization set 360 may be hardcoded and is used to reset and initialize the device. The initialization set 360 includes an idle state 362, a reset command 364, a connect command 366, and a reserved operation 368. The idle state 362 indicates that the device is not performing any meaningful operation and is idle. The reset command 364 causes the device to reset and perform a self-test operation. The connect command 366 sets the connect bit in the status register 240. The reserved operation 368 is to be reserved for other operations or commands or for non-implemented operation. A command that corresponds to the reserved operation 368 results in a "not-supported" error[24].

The attestation set 370 includes a signing operation 372, a public key enumeration 374, and a key certificate enumeration 376. The signing operation 372 provides the remote attestation to verify the validity of the platform running a particular software in the secure environment. The public key enumeration 374 enumerates any additional public keys that are not part of the static configuration information 232. The key certificate enumeration 376 enumerates any additional key certificates that are not part of the static configuration information 232[25].

The signing operation 372 includes a hash function 410 and a cryptographic function 420. The hash function 410 performs hashing on the processor nub loader hash

[22] See Specification, page 16, lines 17-21; Figure 3.
[23] See Specification, page 16, lines 22-24; page 17, lines 1-2.
[24] See Specification, page 17, lines 3-13.
[25] See Specification, page 17, lines 14-20.

272, the chipset hash log 274, the software hash 276, and the nonce 278. The result of this hashing operation is then encrypted by the cryptographic function 420 using the private key 280 stored in the chipset. The result of the encryption becomes the output data 264 to be stored in the IOB 260[26].

The status register 240 includes a self-test field 510, a connection field 520, an estimate field 530, and a reserved field 540. The self-test field 510 provides a result of the self-test operation in response to the reset command. The result may include a failure. When there is a failure, all results from the device are ignored. This failure code is typically reset by a reset command or a system reset. The connection field 520 indicates that the device is responsive to the connect command. The estimate field 530 provides an estimate in some time unit (e.g., milliseconds) to indicate how long a current operation is expected to take. The reserved field 540 is reserved for future use[27].

VI. ISSUES

The issues are:

(1) whether claims 1-5, 21-25, 41-45 and 61-65 are anticipated under 35 U.S.C. §102(e) over U.S. Patent No. 6,357,004 issued to Davis ("Davis-004").

(2) whether claims 6-7, 9-19, 26-27, 29-39, 46-47, 49-59, 66-67 and 69-79 are obvious under 35 U.S.C. §103(x) over Davis-004 in view of U.S. Patent No. 4,319,233 issued to Ermolovich ("Ermolovich").

(3) whether claims 8, 28, 48 and 68 are obvious under 35 U.S.C. §103(a) over Davis-004 in view of Ermolovich as applied to claim 67, and further in view of U.S. Patent No. 5,844,986 issued to Davis ("Davis-986").

VII. GROUPING OF CLAIMS

Applicants contend that the claims of the present invention form into three groups: (1) Group 1 includes claims 1-5, 21-25, 41-45 and 61-65; (2) Group 2 includes claims 6-7, 9-19, 26-27, 29-39, 46-47, 49-59, 66-67 and 69-79; and (3) Group 3 includes claims 8, 28, 48 and 68.

---

[26] See Specification, page 18, lines 5-13.
[27] See Specification, page 18, lines 15-24; Figure 5.

VIII. ARGUMENTS

A. Claims 1-5, 21-25, 41-45 and 61-65 Are Not Anticipated By Davis-004.

Claims 1-5, 21-25, 41-45 and 61-65 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,357,004 issued to Davis ("Davis-004"); Applicants respectfully traverse the rejections and contend that the Examiner has not met the burden of establishing a prima facie case of anticipation.

Applicants contend that Davis-004 does not disclose, expressly or inherently, (1) a secure environment for an isolated execution mode, (2) a processor operating in one of a normal execution mode and the isolated execution mode, (3) an interface to map a device to an address space of a chipset in the secure environment, and (4) a communication storage to allow the device to exchange security information with the processor in the isolated execution mode in a remote attestation.

Davis-004 discloses a system and method for ensuring integrity throughout post-processing. A system includes a host processor and a manipulation processing element (Davis-004, col. 3, lines 58-63). The manipulation processing element comprises a device contained within a package (Davis-004, col. 4, lines 23-25). The device supports post-processing operations and cryptographic operations such as encryption and/or decryption, creation of a digital signature, performance of a hash function, and generation of keys (Davis-004, col. 4, lines 29-37). None of these elements corresponds to an isolated execution mode.

In the Office Action dated November 12, 2003, the Examiner states that Davis-004 discloses one processor having one of a normal execution mode and an isolated execution mode (Office Action, page 4, paragraph 9). Applicants respectfully disagree.

First, Davis-004 discloses two separate processors: a host processor and a manipulation processing element, not a processor having two modes of operations. The memory unit accessible to the manipulation processing element is not accessible to the host processor (Davis-004, col. 4, lines 59-67).

Second, the manipulating processing element may perform post-processing and cryptographic operations, but not operations in an isolated execution mode. As supported in the Specification, the isolated execution mode includes configuration for isolated execution, definition of an isolated area, definition of isolated instructions, generation of

isolated access bus cycles, and generation of isolated mode interrupts (See Specification, page 8, lines 22-25); page 9, lines 1-2). Claims should be interpreted consistently with the specification, which provides content for the proper construction of the claims because it explains the nature of the patentee's invention. See Renishaw, 158 F.3d 1243, 48 USPQ2d 1117 (Fed. Cir. 1998). Here, the meaning of the isolated execution mode must be interpreted consistently with the Specification.

Third, Davis-004 does not disclose communication storage to allow a device to exchange security information with the processor in the isolated execution mode in a remote attestation. Davis-004 merely discloses that for authentication, digital signature may be accompanied by a digital certificate chain (Davis-004, col. 3, lines 14-16). There is no communication storage. The manipulation processing element merely performs post-processing operations on information after the information has been digitally signed (Davis-004, col. 3, lines 38-40). There is no remote attestation.

In the final Office Action dated March 3, 2004, the Examiner states that the features upon which applicants rely are not recited in the rejected claim (s) (Office Action, page 2, paragraph 3). The Examiner then states that "limitations from the specification are not read into the claims", citing In re Van Guens, 988 F.2d. 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Applicants respectfully disagree.

First, the features recited by the Applicants in the previous response are to provide the Examiner an opportunity to review the specification. Those are intended to help interpreting the claim language. They are not limitations read into the claims.

Second, the Examiner apparently misread Van Guens. In Van Guens, the claim in question recites a magnet assembly with a uniform magnetic field. The applicant in Van Guens argues that the uniform magnetic field limitation must be interpreted in light of the specification which discloses the magnetic field uniformity for NMR or MRI imaging. In rejecting this argument, the Court states that the "short answer is that [the] claim ... is not expressly limited to NMR or MRI apparatus". In re Van Guens, 26 USPQ2d at 1059. The court further states that the applicant cannot read an NMR limitation into the claim to justify his argument as to the meaning of the uniform magnetic field. The Van Guens rule, therefore, is only applicable if the claim does not expressly recite the limitation.

Here, the limitation of "an isolated execution mode" is recited in the claim. This limitation should be interpreted according to the specification as discussed above.

Claims should be interpreted consistently with the specification, which provides content for the proper construction of the claims because it explains the nature of the patentee's invention. Renishaw, 158 F.3d 1243, 48 USPQ2d 1117 (Fed. Cir. 1998). The Renishaw court explicitly states that when "a patent applicant has elected to be a lexicographer by providing an explicit definition in the specification for a claim term, .. the definition selected by the applicant controls." Renishaw, 48 USPQ2d 117, 1121. During patent examination, the pending claims must be "given the broadest reasonable interpretation consistent with the specification". See MPEP 2111. "When the applicant states the meaning that the claim terms are intended to have, the claims are examined with that meaning, in order to achieve a complete exploration of the applicant's invention and its relation to the prior art". In re Zletz, 893 F.2d 319, 321, 13 USPQ2d 13320, 1322 (Fed. Cir. 1989). In In re Prater, 415 F.2d 1393, 1404-05, 162 USPQ 541, 550-51 (CCPA 1969) the Court explained that "reading a claim in light of the specification, to thereby interpret limitations explicitly recited in the claim, is a quite different thing from "reading limitations of the specification into a claim," to thereby narrow the scope of the claim by implicitly adding disclosed limitations which have no express basis in the claim". See also In re Morris, 127 F.3d 1048, 1054-55, 44 USPQ 2d 1023, 1027-28 (Fed. Cir. 1997) (The Court held that the PTO is not required, in the course of prosecution, to interpret claims in applications in the same manner as a court would in interpreting claims in an infringement suit. Rather, the "PTO applies to verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in applicant's specification.")

In the final Office Action dated March 3, 2004, the Examiner further states that Davis-4 discloses a communication storage and a remote attestation, citing Davis-4 at Col. 3, lines 15-30 ("..stored within [a] second electronic system [and certification authority]"). The Examiner apparently equates the "communication storage" in the claimed invention with the "second electronic system" in Davis-4. Applicants respectfully disagree.

First, the "second electronic system" in Davis-4 is not a communication storage to exchange security information with the processor in the isolated execution mode. It is merely another system that contains the public key of a certification authority (Davis-4,

col. 3, lines 24-26). Second, the second electronic system does not correspond to the address space as recited in the claim, by virtue of its being a second system separate from the first system associated with the private key. Third, as discussed above, Davis-4 does not disclose an isolated execution mode.

Therefore, Applicants believe that independent claims 1, 21, 41, 61 and their respective dependent claims are distinguishable over the cited prior art references.

B. Claims 6-7, 9-19, 26-27, 29-39, 46-47, 49-59, 66-67 and 69-79 Are Not Obvious over Davis-004 in View of Ermolovich.

Claims 6-7, 9-17, 26-27, 29-37, 46-47, 49-57, 66-67 and 69-77 are rejected under 35 U.S.C. §103(a) as being unpatentable over Davis-004 in view of U.S. Patent No. 4,319,233 issued to Ermolovich ("Ermolovich "). Applicants respectfully traverse the rejection and contend that the Examiner has not met the burden of establishing a prima facie case of obviousness. To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *MPEP §2143, p. 2100-124 (8th Ed., rev. 2, May 2004).* Applicants respectfully contend that there is no suggestion or motivation to combine their teachings, and thus no *prima facie* case of obviousness has been established.

Davis-004 discloses a system and method for ensuring integrity throughout post-processing as discussed above.

Ermolovich discloses a communications device for data processing system. A device status is built and inserted into a packet as a status longword before inserting a command packet into a termination queue (Ermolovich, coo. 85, lines 37-41). The device status contains the status of a communication device after the communication device processes a command packet (Ermolovich, col. 13, lines 37-43). A command interpreter transfers contents of a command field to a command register in an external device (Ermolovich, col. 12, lines 2-6). The communication device may directly write to or read from buffers in the data block and command block (Ermolovich, col. 7, lines 54-58).

Davis-004 and Ermolovich, taken alone or in combination, does not disclose, suggest, or render obvious (1) a secure environment for an isolated execution mode, (2) a processor operating in one of a normal execution mode and the isolated execution mode, (3) an interface to map a device to an address space of a chipset in the secure environment, (4) a communication storage to allow the device to exchange security information with the processor in the isolated execution mode in a remote attestation, (5) a status register to store the status of the device, (6) a command register to store a device command for a command interface set, and (7) an input/output block (IOB) to store input and output data corresponding to the command. There is no motivation to combine Davis-004 and Ermolovich because none of them addresses the problem of isolated execution. There is no teaching or suggestion that a processing having normal and isolated execution modes is present. Davis-004, read as a whole, does not suggest the desirability of a configuration storage in a communication storage corresponding to an address space for an isolated execution mode. Ermolovich merely discloses status word in a command packet for a communication device, not a configuration storage for an isolated execution mode.

In the final Office Action dated March 3, 2004, the Examiner states that Ermolovich discloses a status register to store device status (see col. 85, lines 37-45), a command register to store a device command (see col. 12, lines 2-6) and an input/output block to store input and output data (see col. 71, lines 40-64) (Final Office Action, Page 7, paragraph 7). Applicants respectfully disagree.

Ermolovich merely discloses the communications device determining the status of response to the command that defines a data transfer operation and stores that status in the status word location of the command packet (Ermolovich, col. 14, lines 36-39). The status therefore merely reflects the data transfer operation, not an operation in a remote attestation. Furthermore, it is not contained in a register of an attestation device and is merely inserted into a packet. Regarding the command register, Ermolovich merely discloses transferring the contents of the command field to a command register in an external device (Ermolovich, col. 12, lines 2-6). The command field merely defines commands that effect data transfers such as read, read device chained, write, write device chained, set and clear random enable, no op, set halt, etc. (Ermolovich, col. 12, lines 19-64). None of these commands corresponds to a command interface set including a reset command and a connect command. Regarding the input/output block (IOB), Ermolovich

merely discloses the communications device to directly write to or read from buffers in the data block and the command block (Ermolovich, col. 71, lines 54-59). These blocks are not the input and output data corresponding to the command, including the result of the encryption and the result of the signing operation (See, for example, Specification, page 18, lines 11-13).

In addition, Ermolovich merely discloses a communication device for data processing system. Ermolovich does not disclose or suggest that the communication device operates in a secure environment with the isolated execution mode and in a remote attestation.

Accordingly, claims 6-7, 9-17, 26-27, 29-37, 46-47, 49-57, 66-67 and 69-77 are not obvious over Davis-004 in view Ermolovich.

C. Claims 8, 28, 48 and 68 Are Not Obvious over Davis-004 in View of Ermolovich and Further in View of Davis-986.

Claims 8, 28, 48 and 68 are rejected under 35 U.S.C. §103(a) as being unpatentable over Davis-004 in view of Ermolovich and further in view of U.S. Patent No 5,844,986 issued to Davis ("Davis-986") Applicants respectfully traverse the rejection and contend that the Examiner has not met the burden of establishing a prima facie case of obviousness. To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *MPEP §2143, p. 2100-124 (8th Ed., rev.2, May 2004).* Applicants respectfully contend that there is no suggestion or motivation to combine their teachings, and thus no *prima facie* case of obviousness has been established.

Davis-004 discloses a system and method for ensuring integrity throughout post-processing as discussed above. Ermolovich discloses a communications device for data processing system as discussed above.

Davis-986 discloses a electronic system and method for controlling access through user authentication. In a field BIOS upgrade, a software manufacturer (the BIOS vendor)

sends the user a diskette containing a new BIOS code (Davis-986, col. 3, lines 38-40). An authentication is performed to ensure that the revision date is appropriate (Davis-986, col. 4, lines 7-15).

Davis-004, Ermolovich and Davis-986, taken alone or in any combination, does not disclose, suggest, or render obvious (1) a secure environment for an isolated execution mode, (2) a processor operating in one of a normal execution mode and the isolated execution mode, (3) an interface to map a device to an address space of a chipset in the secure environment, (4) a communication storage to allow the device to exchange security information with the processor in the isolated execution mode in a remote attestation, (5) a configuration storage to store device configuration information, (6) a manufacturer identifier storage to store a manufacturer identifier, and (7) a revision storage to store a revision identifier. There is no motivation to combine Davis-004, Ermolovich and Davis-981 because none of them addresses the problem of isolated execution. There is no teaching or suggestion that a processing having normal and isolated execution modes is present. Davis-004, read as a whole, does not suggest the desirability of a configuration storage in a communication storage corresponding to an address space for an isolated execution mode. Ermolovich merely discloses status word in a command packet for a communication device, not a configuration storage for an isolated execution mode. Davis-986 merely discloses a BIOS upgrade, not a configuration of a device to exchange security information with a processor having normal and isolated execution modes.

In the final Office Action dated March 3, 2004, the Examiner states that Davis-986 discloses manufacturer identifier storage (see col. 3, lines 37-45, software manufacture (BIOS vendor), BIOS code) and a revision storage to store revision identifier (see col. 4, lines 7-13, revision date). Applicants respectfully disagree.

Regarding the manufacturer identifier, Davis-986 merely discloses that "the software manufacturer (the BIOS vendor) will send the user a diskette containing the new BIOS code, and the code to perform the upgrade operations." (Davis-986, Col. 3, lines 38-41). This is not the same as storing the manufacturer identifier in a configuration storage in a secure environment.

Regarding the revision identifier, Davis-986 merely discloses "the digital signature supplied with the 'new BIOS program' may be valid, but the revision date may be inappropriate (e.g., older than the currently installed BIOS)" (Davis-986, col. 4, lines 9-

13). Davis-986, therefore, does not disclose a configuration storage having a revision storage to store a revision identifier. The revision identifier as recited in claims 8, 28, and 48 is to provide a revision number of the attestation device (See specification, page 16, line 23), not a revision date of an update BIOS program as disclosed in Davis-986.

In the present invention, the cited references do not expressly or implicitly suggest a configuration storage in a communication storage corresponding to an address space for an isolated execution mode. In addition, the Examiner failed to present a convincing line of reasoning as to why a combination of Davis-004, Ermolovich and Davis-986 is an obvious application of attestation using isolated execution mode.

Accordingly, claims 8, 28, and 48 are not obvious over Davis-004 in view Ermolovich, and further in view of Davis-986.

IX. CONCLUSION

The Examiner failed to establish a prima facie case of anticipation in rejecting claims 1-5, 21-25, 41-45 and 61-65. The Federal Circuit stated that to anticipate a claim, the reference must teach every element of the claim. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Vergegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ...claim." Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989. Here, the cited prior reference does not disclose, expressly or inherently, (1) a secure environment for an isolated execution mode, (2) a processor operating in one of a normal execution mode and the isolated execution mode, (3) an interface to map a device to an address space of a chipset in the secure environment, and (4) a communication storage to allow the device to exchange security information with the processor in the isolated execution mode in a remote attestation.

In addition, the Examiner failed to establish a prima facie case of obviousness and failed to show there is teaching, suggestion or motivation to combine the references in rejecting claims 6-7, 9-17, 26-27, 29-37, 46-47, 49-57, 66-67 and 69-77, and claims 8, 28, and 48. "When determining the patentability of a claimed invention which combined two known elements, 'the question is whether there is something in the prior art as a whole suggest the desirability, and thus the obviousness, of making the combination.'" In re Beattie, Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co., 730 F.2d 1452, 1462, 221 USPQ (BNA) 481, 488 (Fed. Cir. 1984). "To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or implicitly suggest the claimed invention or the Examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references." Ex parte Clapp, 227 USPQ 972, 973. (Bd.Pat.App.&Inter. 1985). In the present invention, the cited references do not expressly or implicitly, disclose, suggest, or render obvious (1) a secure environment for an isolated execution mode, (2) a processor operating in one of a normal execution mode and the isolated execution mode, (3) an interface to map a device to an

address space of a chipset in the secure environment, (4) a communication storage to allow the device to exchange security information with the processor in the isolated execution mode in a remote attestation, (5) a status register to store the status of the device, (6) a command register to store a device command for a command interface set, (7) an input/output block (IOB) to store input and output data corresponding to the command, (8) a manufacturer identifier storage to store a manufacturer identifier, and (9) a revision storage to store a revision identifier. In addition, the Examiner failed to present a convincing line of reasoning as to why a combination of Davis-004, Ermolovich and Davis-986 is an obvious application of attestation using isolated execution mode.

Applicants respectfully request that the Board enter a decision overturning the Examiner's rejection of all pending claims, and holding that the claims are neither anticipated or rendered obvious by the prior art.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: July 27, 2004

THINH V. NGUYEN
Reg. No. 42,034

12400 Wilshire Blvd., 7th Floor
Los Angeles, CA 90025-1026
(714) 557-3800

X. APPENDIX

The claims of the present application which are involved in this appeal are as follows:

1. (original) An apparatus comprising:

an interface to map a device via a bus to an address space of a chipset in a secure environment for an isolated execution mode, the secure environment being associated with an isolated memory area accessible by at least one processor, the at least one processor operating in one of a normal execution mode and the isolated execution mode; and

a communication storage corresponding to the address space to allow the device to exchange security information with the at least one processor in the isolated execution mode in a remote attestation.

2. (original) The apparatus of claim 1 wherein the security information includes at least one of a static public key and a static key certificate.

3. (original) The apparatus of claim 2 wherein the interface comprises:

a decoder to decode the address space onto the bus so that an access to the chipset is passed to the device.

4. (original) The apparatus of claim 3 wherein the device accesses a chipset storage via the address space.

5. (original) The apparatus of claim 4 wherein the communication storage comprises:

a configuration storage to store device configuration information.

6. (original) The apparatus of claim 5 wherein the communication storage further comprises:

a status register to store device status of the device;

a command register to store a device command for a command interface set; and

an input/output block (IOB) to store input and output data corresponding to the command.

7. (original) The apparatus of claim 6 wherein the configuration storage comprises:

a public key storage to store the static public key;

a key certificate storage to store the static key certificate; and

an interface set storage to store an interface set identifier, the interface set identifier identifying a command interface set supported by the device.

8. (original) The apparatus of claim 7 wherein the configuration storage further comprises:

a manufacturer identifier storage to store a manufacturer identifier; and

a revision storage to store a revision identifier.

9. (original) The apparatus of claim 7 wherein the command interface set is an initialization set, the initialization set supporting a reset command and a connect command.

10. (original) The apparatus of claim 7 wherein the command interface set is an attestation set, the attestation set performing at least one of a public key enumeration, a key certificate enumeration, and a signing operation.

11. (original) The apparatus of claim 10 wherein the status register comprises:

a connection field to provide a connection status to indicate that the device is responsive to the connect command; and

an estimate field to provide an estimate of processing time for an operation specified in the command.

12. (original) The apparatus of claim 11 wherein the status register further comprises:

a self-test field to indicate status of a self test in response to the reset command.

13. (original) The apparatus of claim 10 wherein the public key enumeration enumerates an additional public key other than the static public key.

14. (original) The apparatus of claim 10 wherein the key certificate enumeration enumerates an additional key certificate other than the static key certificate.

15. (original) The apparatus of claim 10 wherein the sign operation generates a signature to attest validity of the secure environment using a private key provided by the chipset.

16. (original) The apparatus of claim 15 wherein the signature corresponds to signing a chipset parameter.

17. (previously presented) The apparatus of claim 16 wherein the chipset parameter is one of a processor nub loader hash, a chipset hash log, a software hash, and a nonce.

18. (previously presented) The apparatus of claim 17 wherein the processor nub loader hash and the chipset hash log are stored in the chipset storage.

19. (previously presented) The apparatus of claim 18 wherein the software hash and the nonce are provided by a processor nub.

20. (original) The apparatus of claim 19 wherein the output data include the signature.

21. (original) A method comprising:

mapping a device via a bus to an address space of a chipset in a secure environment for an isolated execution mode, the secure environment being associated with an isolated memory area accessible by at least one processor, the at least one processor operating in one of a normal execution mode and the isolated execution mode; and

exchanging security information between the device and the at least one processor in the isolated execution mode in a remote attestation via a communication storage corresponding to the address space.

22. (original) The method of claim 21 wherein the security information includes at least one of a static public key and a static key certificate.

23. (original) The method of claim 22 wherein mapping comprises:

decoding the address space onto the bus so that an access to the chipset is passed to the device.

24. (original) The method of claim 23 wherein the device accesses a chipset storage via the address space.

25. (original) The method of claim 24 wherein exchanging comprises:

storing device configuration information in a configuration storage.

26. (original) The method of claim 25 wherein exchanging further comprises:

storing device status of the device in a status register;

performing a device command corresponding to a command interface set to a command register; and

storing input and output data corresponding to the command in an input/output block (IOB).

27. (original) The method of claim 26 wherein storing in the configuration storage comprises:

storing the static public key in a public key storage;

storing the static key certificate in a key certificate storage; and

storing an interface set identifier in an interface set storage, the interface set identifier identifying a command interface set supported by the device.

28. (original) The method of claim 27 wherein storing in the configuration storage further comprises:

storing a manufacturer identifier in a manufacturer identifier storage; and

storing a revision identifier in a revision storage.

29. (original) The method of claim 27 wherein performing the device command comprises performing a reset command and a connect command corresponding to an initialization set.

30. (original) The method of claim 27 wherein performing the device command comprises performing at least one of a public key enumeration, a key certificate enumeration, and a signing operation, the public key enumeration, the key certificate enumeration, and the signing operation corresponding to an attestation set.

31. (original) The method of claim 30 wherein storing the device status comprises:

providing a connection status to indicate that the device is responsive to the connect command; and

providing an estimate of processing time for an operation specified in the command.

32. (original) The method of claim 31 wherein storing the device status further comprises:

indicating status of a self test in response to the reset command.

33. (original) The method of claim 30 wherein performing the public key enumeration comprises enumerating an additional public key other than the static public key.

34. (original) The method of claim 30 wherein performing the key certificate enumeration comprises enumerating an additional key certificate other than the static key certificate.

35. (original) The method of claim 30 wherein performing the sign operation comprises generating a signature to attest validity of the secure environment using a private key provided by the chipset.

36. (original) The method of claim 35 wherein the signature corresponds to signing a chipset parameter.

37. (previously presented) The method of claim 36 wherein the chipset parameter is one of a processor nub loader hash, a chipset hash log, a software hash, and a nonce.

38. (previously presented) The method of claim 37 wherein the processor nub loader hash and the chipset hash log are stored in the chipset storage.

39. (previously presented) The method of claim 38 wherein the software hash and the nonce are provided by a processor nub.

40. (original) The method of claim 39 wherein the output data include the signature.

41. (original) A computer program product comprising:

a machine readable medium having program code embedded therein, the computer program product comprising:

computer readable program code for mapping a device via a bus to an address space of a chipset in a secure environment for an isolated execution mode, the secure environment being associated with an isolated memory area accessible by at least one processor, the at least one processor operating in one of a normal execution mode and the isolated execution mode; and

computer readable program code for exchanging security information between the device and the at least one processor in the isolated execution mode in a remote attestation via a communication storage corresponding to the address space.

42. (original) The computer program product of claim 41 wherein the security information includes at least one of a static public key and a static key certificate.

43. (original) The computer program product of claim 42 wherein the computer readable program code for mapping comprises:

computer readable program code for decoding the address space onto the bus so that an access to the chipset is passed to the device.

44. (original) The computer program product of claim 43 wherein the device accesses a chipset storage via the address space.

45. (original) The computer program product of claim 44 wherein the computer readable program code for exchanging comprises:

computer readable program code for storing device configuration information in a configuration storage.

46. (original) The computer program product of claim 45 wherein the computer readable program code for exchanging further comprises:

computer readable program code for storing device status of the device in a status register;

computer readable program code for performing a device command corresponding to a command interface set to a command register; and

computer readable program code for storing input and output data corresponding to the command in an input/output block (IOB).

47. (original) The computer program product of claim 46 wherein the computer readable program code for storing in the configuration storage comprises:

computer readable program code for storing the static public key in a public key storage;

computer readable program code for storing the static key certificate in a key certificate storage; and

computer readable program code for storing an interface set identifier in an interface set storage, the interface set identifier identifying a command interface set supported by the device.

48. (original) The computer program product of claim 47 wherein the computer readable program code for storing in the configuration storage further comprises:

computer readable program code for storing a manufacturer identifier in a manufacturer identifier storage; and

computer readable program code for storing a revision identifier in a revision storage.

49. (original) The computer program product of claim 47 wherein the computer readable program code for performing the device command comprises performing a reset command and a connect command corresponding to an initialization set.

50. (original) The computer program product of claim 47 wherein the computer readable program code for performing the device command comprises performing at least one of a public key enumeration, a key certificate enumeration, and a signing operation, the public key enumeration, the key certificate enumeration, and the signing operation corresponding to an attestation set.

51. (original) The computer program product of claim 50 wherein the computer readable program code for storing the device status comprises:

computer readable program code for providing a connection status to indicate that the device is responsive to the connect command; and

computer readable program code for providing an estimate of processing time for an operation specified in the command.

52. (original) The computer program product of claim 51 wherein the computer readable program code for storing the device status further comprises:

computer readable program code for indicating status of a self test in response to the reset command.

53. (original) The computer program product of claim 50 wherein the computer readable program code for performing the public key enumeration comprises enumerating an additional public key other than the static public key.

54. (original) The computer program product of claim 50 wherein the computer readable program code for performing the key certificate enumeration comprises enumerating an additional key certificate other than the static key certificate.

55. (original) The computer program product of claim 50 wherein the computer readable program code for performing the sign operation comprises generating a signature to attest validity of the secure environment using a private key provided by the chipset.

56. (original) The computer program product of claim 55 wherein the signature corresponds to signing a chipset parameter.

57. (previously presented) The computer program product of claim 56 wherein the chipset parameter is one of a processor nub loader hash, a chipset hash log, a software hash, and a nonce.

58. (previously presented) The computer program product of claim 57 wherein the processor nub loader hash and the chipset hash log are stored in the chipset storage.

59. (previously presented) The computer program product of claim 58 wherein the software hash and the nonce are provided by a processor nub.

60. (original) The computer program product of claim 59 wherein the output data include the signature.

61. (original) A system comprising:

at least one processor operating in a secure environment, the at least one processor having one of a normal execution mode and an isolated execution mode;

a memory coupled to the at least one processor, the memory having an isolated memory area accessible to the at least one processor in the isolated execution mode; and

a chipset coupled to the at least one processor and the memory, the chipset having a circuit, the circuit comprising:

an interface to map a device via a bus to an address space of the chipset in the secure environment, and

a communication storage corresponding to the address space to allow the device to exchange security information with the at least one processor in the isolated execution mode in a remote attestation.

62. (original) The system of claim 61 wherein the security information includes at least one of a static public key and a static key certificate.

63. (original) The system of claim 62 wherein the interface comprises:

a decoder to decode the address space onto the bus so that an access to the chipset is passed to the device.

64. (original) The system of claim 63 wherein the device accesses a chipset storage via the address space.

65. (original) The system of claim 64 wherein the communication storage comprises:

a configuration storage to store device configuration information.

66. (original) The system of claim 65 wherein the communication storage further comprises:

a status register to store device status of the device;

a command register to store a device command for a command interface set; and

an input/output block (IOB) to store input and output data corresponding to the command.

67. (original) The system of claim 66 wherein the configuration storage comprises:

a public key storage to store the static public key;

a key certificate storage to store the static key certificate; and

an interface set storage to store an interface set identifier, the interface set identifier identifying a command interface set supported by the device.

68. (original) The system of claim 67 wherein the configuration storage further comprises:

a manufacturer identifier storage to store a manufacturer identifier; and

a revision storage to store a revision identifier.

69. (original) The system of claim 67 wherein the command interface set is an initialization set, the initialization set supporting a reset command and a connect command.

70. (original) The system of claim 67 wherein the command interface set is an attestation set, the attestation set performing at least one of a public key enumeration, a key certificate enumeration, and a signing operation.

71. (original) The system of claim 70 wherein the status register comprises:

a connection field to provide a connection status to indicate that the device is responsive to the connect command; and

an estimate field to provide an estimate of processing time for an operation specified in the command.

72. (original) The system of claim 71 wherein the status register further comprises:

a self-test field to indicate status of a self test in response to the reset command.

73. (original) The system of claim 70 wherein the public key enumeration enumerates an additional public key other than the static public key.

74. (original) The system of claim 70 wherein the key certificate enumeration enumerates an additional key certificate other than the static key certificate.

75. (original) The system of claim 70 wherein the sign operation generates a signature to attest validity of the secure environment using a private key provided by the chipset.

76. (original) The system of claim 75 wherein the signature corresponds to signing a chipset parameter.

77. (previously presented) The system of claim 76 wherein the chipset parameter is one of a processor nub loader hash, a chipset hash log, a software hash, and a nonce.

78. (previously presented) The system of claim 77 wherein the processor nub loader hash and the chipset hash log are stored in the chipset storage.

79. (previously presented) The system of claim 78 wherein the software hash and the nonce are provided by a processor nub.

80. (original) The system of claim 79 wherein the output data include the signature.

AF 1120

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application No.	09/541,667
Filing Date	March 31, 2000
First Named Inventor	Carl M. Ellison
Art Unit	2134
Examiner Name	Tongoc Tran
Total Number of Pages in This Submission	102
Attorney Docket Number	42390P8629

ENCLOSURES *(check all that apply)*

- ☒ Fee Transmittal Form
  - ☒ Fee Attached
- ☐ Amendment / Response
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
  - ☐ PTO/SB/08
- ☐ Certified Copy of Priority Document(s)
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Basic Filing Fee
  - ☐ Declaration/POA
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s)
- ☐ After Allowance Communication to Group
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☒ Appeal Communication to Group **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☒ Other Enclosure(s) *(please identify below):*

2 copies of Appeal Brief

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Thinh V. Nguyen, Reg. No. 42,034 BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Signature	
Date	July 27, 2004

CERTIFICATE OF MAILING/TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal , Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Typed or printed name	**Tu T. Nguyen**		
Signature		Date	**July 27, 2004**

FEE TRANSMITTAL for FY 2004

Effective 01/01/2004. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27.

TOTAL AMOUNT OF PAYMENT ($) 330.00

Complete if Known	
Application Number	09/541,667
Filing Date	March 31, 2000
First Named Inventor	Carl M. Ellison
Examiner Name	Tongoc Tran
Art Unit	2134
Attorney Docket No.	42390P8629

METHOD OF PAYMENT *(check all that apply)*

☒ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☐ Deposit Account

Deposit Account Number: 02-2666

Deposit Account Name: Blakely, Sokoloff, Taylor & Zafman LLP

The Commissioner is authorized to: *(check all that apply)*

☐ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s) or underpayment of fees as required under 37 CFR §§ 1.16, 1.17, 1.18 and 1.20.

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($)

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below	Fee Paid
Total Claims		- 20** =		X		=
Independent Claims		- 3 =		X		=
Multiple Dependent						=

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple Dependent claim, if not paid
1204	86	2204	43	**Reissue independent claims over original patent
1205	18	2205	9	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)

***or number previously paid, if greater, For Reissues, see below*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
2053	130	2053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1404	330	2401	165	Notice of Appeal	330.00
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	2451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	2460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	1809	385	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	
Other fee (specify)					

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 330.00

SUBMITTED BY

Complete (if applicable)

Name *(Print/Type)*	Thinh V. Nguyen	*Registration No. (Attorney/Agent)*	42,034	*Telephone*	(714) 557-3800
Signature				*Date*	07/27/04

Based on PTO/SB/17 (10-03) as modified by Blakely, Sokoloff, Taylor & Zafman (wlr) 02/10/2004.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450